Installation and Upgrade 1

Installing DataPlane

Date of Publish: 2020-03-26

https://docs.hortonworks.com

Contents

DataPlane Platform support requirements

You should review the support requirements for the DP Platform to ensure your environment meets those requirements. Additionally, you must consider various aspects of your clusters and prepare those clusters as part of your DataPlane installation in order to register the clusters with DP Platform.

⚠ **Important:**

The specific DataPlane Apps you plan to install into your environment might bring additional requirements. Review the App-specific documentation thoroughly to ensure you can meet the App-specific requirements. For example, depending on your choice of Apps, your cluster requirements might change. This includes (but is not limited to) a minimal HDP version, setup and configuration of Knox, and other cluster requirements.

Support Matrix information

You can find the most current information about interoperability for this release on the Support Matrix. The Support Matrix tool provides information about:

- Operating Systems
- Databases
- Browsers
- JDKs

To access the tool, go to: https://supportmatrix.hortonworks.com.

DP Platform Host requirements

DP Platform must be installed on a dedicated host that is not part of an existing HDP cluster, to prevent potential port conflicts with other cluster services.

Table 1: Requirements for the DP Platform host

Item	Versions
Operating Systems	https://supportmatrix.hortonworks.com
Databases	https://supportmatrix.hortonworks.com
Container infrastructure	Docker 18.x or higher
Processing and Memory Requirements	• Multicore processor, with minimum 8 cores • Minimum 16 GB RAM
Software	• yum, rpm • wget • tar • bash shell
Authentication	Existing LDAP or Active Directory (AD)
Environment	• Disable SELinux • Firewall and IP Table rules to allow Docker network communication and DP communication as per ports below
Ports	• 443 (Default port used by DataPlane for SSL access) • 80 (Redirected to port 443 for SSL) • 8500 (Default port used by Consul which handles the Docker container networking)

Related Tasks

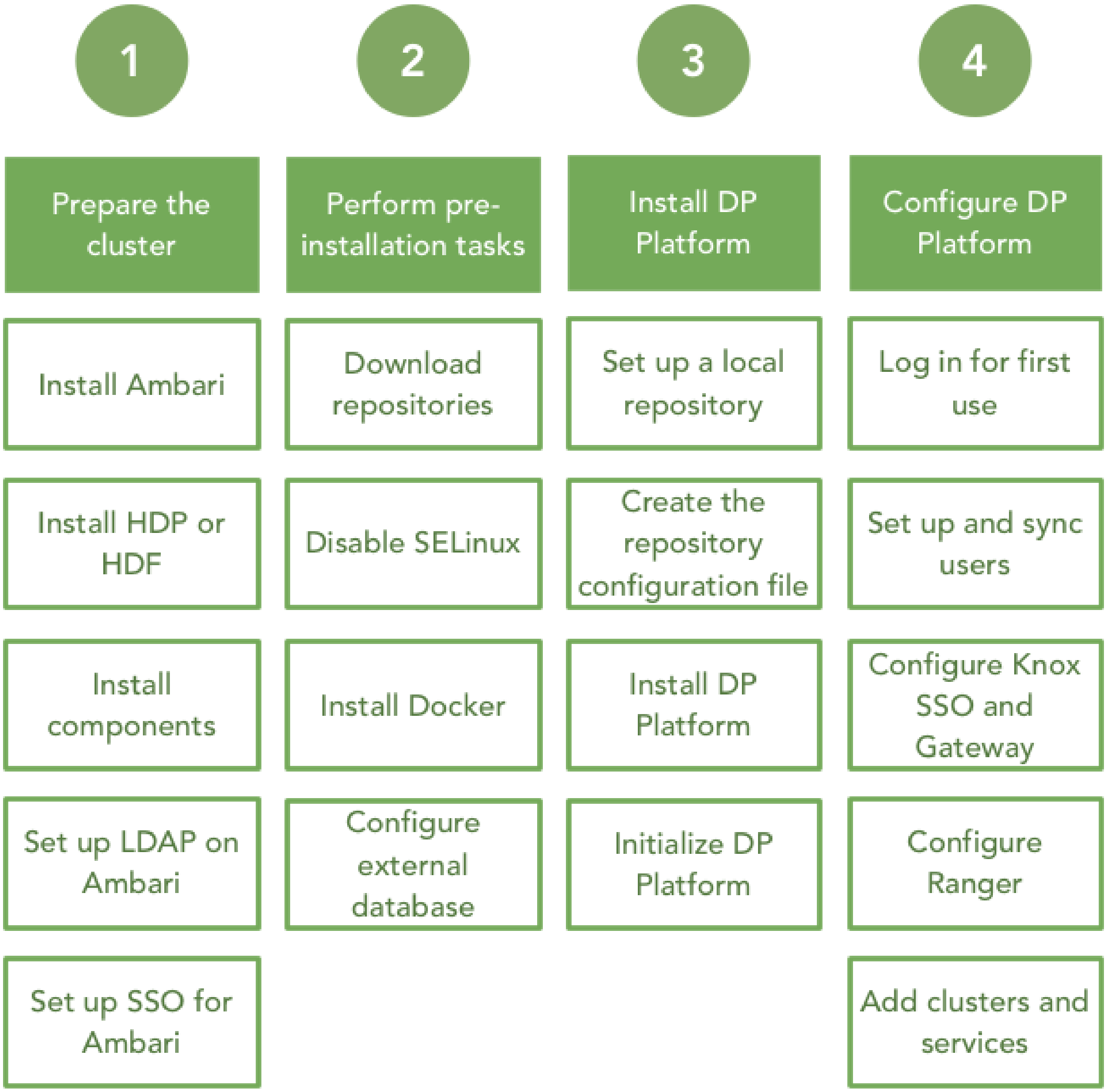
General requirements for clusters

Installation overview

DP Platform and its associated DP Apps are installed on a single host. The DP Platform and the DP Apps run as a set of “containers” on Docker on this host. It is recommended to have this as a dedicated host distinct from other software or cluster hosts. Let us refer to this host as your DP Instance (or “DP Host”).

Hosts from clusters that you plan to register into DataPlane must be accessible from this host. The hostname of a cluster node must be DNS addressable from the DataPlane host. In addition, for any DP Apps you plan to use with these clusters, you must install the requisite Cluster Agent for that DP App (for example: DLM Engine or DSS Profiler). Be sure your clusters meet the hardware and software requirements for that particular Agent. See the DataPlane Support Matrix and the support matrix for each of the DP Apps that you want to install.

You are strongly encouraged to read completely through this entire document before starting the installation process, so that you understand the interdependencies and order of the steps.



1	2	3	4
Prepare the cluster	Perform pre-installation tasks	Install DP Platform	Configure DP Platform
Install Ambari	Download repositories	Set up a local repository	Log in for first use
Install HDP or HDF	Disable SELinux	Create the repository configuration file	Set up and sync users
Install components	Install Docker	Install DP Platform	Configure Knox SSO and Gateway
Set up LDAP on Ambari	Configure external database	Initialize DP Platform	Configure Ranger
Set up SSO for Ambari			Add clusters and services

Pre-installation tasks

Prepare your cluster by installing or upgrading to the required version of HDP or HDF and make sure you perform all the other prerequisite tasks.

Procedure

1. Prepare your clusters.
2. Perform the pre-installation tasks.

Prepare your clusters

Make sure you complete these basic preparation steps for any cluster you plan to use with DataPlane.

1. Install or upgrade to the supported version of Ambari. See Apache Ambari Installation for more details. See Support Matrix for details of the supported Ambari versions.
2. Install or upgrade to the supported versions of HDP or HDF on your cluster using Ambari. See the Ambari Installation documentation for more details. See DP App specific Support Matrix for details of the supported HDP and HDF versions.
3. Install and configure the required cluster services in your cluster.Make sure the DP App-specific components are installed in the cluster. For each app-specific list of requirements, see the Product Interoperability details in the apps' installation documentation.
4. Set up LDAP Authentication for Ambari.
    a. For Ambari 2.7: See Configuring Ambari Authentication with LDAP/AD.
    b. For Ambari 2.6: See Configuring Ambari for LDAP Authentication.
5. Set up Knox Authentication. See Knox Authentication for DataPlane Clusters in DataPlane Getting Started for more information.
    a. Knox SSO

        Note: Use the following information to register the cluster in DataPlane. Currently, DataPlane does not allow changing the value of cookie to anything other than hadoop-jwt. This value is used internally by DataPlane.

        <param><name>knox.token.client.data</name>

        <value>cookie.name=hadoop-jwt</value></param>

        1. For Ambari 2.7: See Configuring Knox SSO.
        2. For Ambari 2.6: See Setting up Knox SSO for Ambari.
    b. Knox Trusted Proxy Pattern - For Ambari 2.7: See Configure Knox TPP for DataPlane for more information.

Prepare your clusters

Make sure you complete these basic preparation steps for any cluster you plan to use with DataPlane.

1. Install or upgrade to the supported version of Ambari. See Apache Ambari Installation for more details. See Support Matrix for details of the supported Ambari versions.
2. Install or upgrade to the supported versions of HDP or HDF on your cluster using Ambari. See the Ambari Installation documentation for more details. See DP App specific Support Matrix for details of the supported HDP and HDF versions.
3. Install and configure the required cluster services in your cluster.Make sure the DP App-specific components are installed in the cluster. For each app-specific list of requirements, see the Product Interoperability details in the apps' installation documentation.
4. Set up LDAP Authentication for Ambari.

   a. For Ambari 2.7: See Configuring Ambari Authentication with LDAP/AD.
   b. For Ambari 2.6: See Configuring Ambari for LDAP Authentication.
5. Set up Knox Authentication. See Knox Authentication for DataPlane Clusters in DataPlane Getting Started for more information.

   a. Knox SSO

      Note: Use the following information to register the cluster in DataPlane. Currently, DataPlane does not allow changing the value of cookie to anything other than hadoop-jwt. This value is used internally by DataPlane.

      <param><name>knox.token.client.data</name>

      <value>cookie.name=hadoop-jwt</value></param>

      1. For Ambari 2.7: See Configuring Knox SSO.
      2. For Ambari 2.6: See Setting up Knox SSO for Ambari.
   b. Knox Trusted Proxy Pattern - For Ambari 2.7: See Configure Knox TPP for DataPlane for more information.

Perform the pre-installation tasks

Complete the pre-installation tasks before you install DP Platform. Refer to the *DataPlane Support Matrix* for requirements and supported databases.

Download product binaries

Download the DataPlane repository tarballs (or the product binaries) from the Hortonworks Customer Portal following the instructions provided as part of the subscription fulfillment process. DP Platform and the DP Apps (and related Cluster Agents) are provided as RPMs in tarball repositories.

Check DNS

Your system must be configured for both forward and reverse DNS.

Every host name used with DataPlane must be resolvable by DNS or configured in the /etc/hosts file on the DataPlane container, so that host names can be resolved between all cluster nodes. Using a DNS server is the recommended method, but if the instances are added to /etc/hosts, you must explicitly register the cluster host names within the DataPlane Docker containers. It is not sufficient to have the host names included in the /etc/hosts file on the DP Platform host. See the DP Platform Administration guide for instructions.

Note: If you are using AWS, do not use the public DNS to access DataPlane. Use a public IP address or set up and use a DNS (Route 53) fully qualified domain name (FQDN).

Check ports

The DP Host requires the following ports to be available on the host and accessible via any configured firewall rules:

Port	Description
80, 443	The DP Instance Web UI for end-user access. Port 80 is redirected to port 443 for SSL. By default, DataPlane configures a self-signed SSL certificate. Refer to *Configuring DP Platform* to configure your own certificate. You can change the default ports. See *DataPlane Administration* for the procedure to change the details of ports.
8500	Used by Consul which handles the Docker container networking. This port is used internally in the DP instance deployment and does not need to be end-user accessible.

Disable SELinux

You must disable SELinux enforcement of permissions before installing DataPlane.

If you do not disable SELinux, then DataPlane will not install and run properly, and you will have to destroy and reinstall the containers.

```
setenforce 0    #A zero, not a letter
sed -i 's/^SELINUX=.*/SELINUX=disabled/g' /etc/sysconfig/selinux
```

Important: The second command prevents SELinux from being automatically re-enabled after a reboot.

Install Docker

Docker containers are used to install DataPlane. You must install either Docker Enterprise Edition (EE) or Community Edition (CE). You might be required to reboot your system after installing Docker.

For general information about installing Docker, see Install Docker.

For Docker installation instructions for your operating system, access the appropriate Docker instructions:

- Get Docker EE for Red Hat Enterprise Linux
- Get Docker EE for CentOS
- Get Docker EE for Oracle Linux
- Get Docker CE for CentOS

Make sure you start the Docker service after installation using the following command:

service docker start

Enable IPv6 module at Kernel level

It is recommended that IPv6 module is enabled at the Kernel level and IPv6 is bound to the default ethernet adapter. This ensures the DataPlane NGinx service is reloaded correctly as required by the service.

Configure LDAP

You need access to an enterprise LDAP setup when configuring DataPlane. Refer to *Enterprise LDAP requirements* for more information on the LDAP settings and options.

Configure external database

By default, DP Platform includes PostgreSQL and MySQL databases. Note the embedded PostgreSQL instance is available for testing and evaluation purposes only.

Note: You should configure your DP instance to use an external PostgreSQL or MySQL instance. We strongly recommend configuring DP Platform with an existing external database when running in production, and not use the embedded PostgreSQL. MySQL database is not supported as an Embedded Database. Refer to *Configuring DP Platform* for more information.

Related Tasks

Advanced: Add host entries to the DataPlane environment

Related reference

DataPlane Support Matrix

Enterprise LDAP requirements

Related Information

DataPlane Administration

Setting up the local repository for DataPlane

You must download the DataPlane repository tarballs from the Hortonworks Customer Portal following the instructions provided as part of the subscription fulfillment process. DP Platform and the DP Apps (and related Cluster Agents) are provided as RPMs in tarball repositories.

Before installing DataPlane, you must set up a server to host the RPMs in a local repository that can then be used to install the product binaries.

Important: You can create the local repository on the same host as your DP Instance, but do not host the local repository on port 80 since that will conflict with your DP Instance.

Prepare the web server for the local repository

Before setting up your local repository, you must properly configure an HTTP web server, on which you will create the repositories.

Before you begin

Prior to preparing the web server, you must have:

- Selected a server host that runs a supported operating system. This will be the local repository host.
- Enabled network access from your target DP Instance host to local repository host.
- Ensured that the web server is not using port 80 if you are using the DP Host for the HTTP web server. This port is used by DataPlane and will cause a conflict if in use by your web server.
- Ensured that the hosts have a package manager installed such as yum (for RHEL, CentOS, or Oracle Linux).

Procedure

1. Create an HTTP server:
   a) On the local repository host, install an HTTP server (such as Apache httpd) using the instructions provided on the Apache community website.
   b) Activate the server.
   c) Ensure that any firewall settings allow inbound HTTP access from your cluster nodes to your local repository host.

   Note:

   If you are using Amazon EC2, make sure that SELinux is disabled.
2. On your local repository host, create a directory for your web server.

   ```
   mkdir -p /var/www/html/
   ```
3. Optional: If you are using a symlink, enable the followsymlinks on your web server.

What to do next

You next must set up your local repository.

Related Information

Downloading the Apache HTTP Server

Set up a local repository for DataPlane

Setting up a local repository involves moving the tarball to the selected mirror server and extracting the tarball to create the repository.

Before you begin

- Ensure that you have downloaded the required tarball from the customer portal, following the instructions provided as part of the product procurement process.
- You must have completed the preparatory tasks before setting up a repository.

Procedure

1. Copy the repository tarballs to the web server directory and expand (uncompress) the archive file:

   a) Navigate to the web server directory you previously created.

   ```
   cd /var/www/html/
   ```

   All content in this directory is served by the web server.

   b) Move the tarball to the current directory and expand each of the repository tarball.

   Replace <filename> with the actual name of the RPM tarball that you are expanding.

   ```
   tar zxvf <filename>.tar.gz
   ```

   During expansion of the tarball, subdirectories are created in /var/www/html/, such as DP/centos7. These directories contain the repositories.

   Expanding the tarball takes several seconds.

2. Confirm that you can browse to the newly created local repositories by using the *Base URL*:

   ```
   http://<your_webserver>:port/<repo_name>/<OS>/<version>
   ```

   - <your_webserver>:port

     This is the FQDN and port of the web server host.
   - <repo_name>

     The repository name, usually the abbreviated name of the DataPlane component, for example DP for *DP Platform*.
   - <OS>

     The operating system.
   - <version>

     The version number of the downloaded component.

   DP Platform Base URL example:

   ```
   http://<your_webserver>:port/DP/centos7/1.2.1.0
   ```

   Remember this base URL. You need it to set up the repository configuration file in subsequent steps.

3. If you have multiple repositories configured in your environment, deploy the following plugin on the nodes in which you want to install DP software - DP Platform, the apps, and the agents on the clusters.

   ```
   yum install yum-plugin-priorities
   ```

4. Edit the /etc/yum/pluginconf.d/priorities.conf file to add the following values:

   ```
   [main]
   enabled=1
   gpgcheck=0
   ```

Results

The local repository is now set up and ready for use.

What to do next

Create the repository configuration file for the newly created local repository.

Create the repository configuration file

A repository configuration file (.repo file) must be created on the DP host. The file is required to identify the path to the repository data, establish whether a GPG signature check should be performed on the repository packages, etc. Only one repository configuration file is needed.

Procedure

1. Navigate to the repository directory.
   cd /etc/yum.repos.d/
2. Create a repository file.
   vi dp.repo

   Alternatively, you can copy an existing repository file to edit.
3. Add the following content to the repository file.

   ⚠ **Important:** Be sure to use the *Base URL* you created when setting up the local repository.

   ```
   #VERSION_NUMBER=1.2.1.0
   [DP-1.2.0.0-xx]
   name=DP  Version - DP-1.2.1.0
   baseurl=http://<your_webserver>:port/DP/centos7/1.2.1.0
   gpgcheck=1
   gpgkey=http://<your_webserver>:port/DP/centos7/1.2.1.0/RPM-GPG-KEY/RPM-
   GPG-KEY-Jenkins
   enabled=1
   priority=1
   ```

Results

You are now ready to install the DataPlane software.

Related Tasks

Install DP Platform

Install and Configure DataPlane

After setting up the local repository, install DP platform. You must configure an external database and then initialize the DP platform.

Procedure

1. Install DP Platform.
2. Configure an external database.
3. Configure a TLS certificate.
4. Initialize DP Platform.
5. Log in and configure DP Platform.

Install DP Platform

When installing DataPlane in a production environment, DP Platform and all associated DP apps must be installed on a separate host that is not part of any cluster.

About this task

DP Platform is required as a baseline, but you can install any combination of DP apps on top of the platform. After installing the platform, see the installation instructions for each DP app for instructions on how to install and configure the app on the platform.

- You will be installing the DP Platform software using the local repositories.

Before you begin

- You need root user access on the DP Host to perform this task.
- You must have completed the actions identified in *DataPlane installation prerequisites*.
- The host must meet the requirements identified in the *DataPlane Support Matrix*.
- The host must be a dedicated host that is not part of an existing cluster. This prevents potential port conflicts with other cluster services.
- You must have the FQDN or IP address of the host available.
- You will be initializing the DP Platform software, and optionally configuring an external database or your own SSL certificate.
- If you plan to use an external database, refer to the DataPlane Support Matrix for requirements and supported databases. Be sure to have that database URI, username and password available and to perform the steps in Configure External Database prior to performing the install steps below. We strongly recommend for any production deployment that you configure an external database.
- If you plan to use an SSL certificate, have the full path and file name for the public key and private key (.pem files) and the certificate password available. Be sure to perform the steps in Configure a TLS Certificate prior to performing the install steps below.

Procedure

1. Log in as root to the host on which you set up the DataPlane repositories.

   ```
   sudo su
   ```

2. Verify that SELinux is enabled in the permissive mode.

   ```
   getenforce
   ```

   If SELinux is not disabled, stop now and verify that all installation prerequisites were successfully completed, then continue with the installation.
3. Be sure Docker is installed, configured, and started on the host.

   Refer to *Installation Prerequisites* for more information.

4. Install the DP Platform.

   ```
   yum install dp-core
   ```

   A folder is created that contains the Docker image tarball files and a configuration script.

   If the yum command fails, then the local repository was not set up correctly. Check the repository file /etc/yum.repos.d/dp.repo on the DP host.

What to do next

Proceed to initializing the DP Platform.

Related Tasks

Perform the pre-installation tasks

Setting up the local repository for DataPlane

Related Information

Hortonworks Support Matrix

Installing DataPlane Platform in SELinux Security Module

Generally, when you install the Dataplane Platform, the SELinux module should be disabled.

About this task

In certain specific cases, where it is not possible to disable SELinux, follow these steps to install DataPlane Platform:

Procedure

1. You must download the **Docker 18+** version.
2. Verify if the SELinux module is enabled by using the command: getenforce

   SELinux must still be enabled.

   The generated output displays:

   ```
   Enforcing
   ```
3. Start the Docker service and check for the status using the command: systemctl status docker

   Make sure that Docker is in Active status and check for the same using the command:docker ps

   The generated output displays the following:

   ```
   CONTAINER ID  IMAGE  COMMAND  CREATED  STATUS  PORTS  NAMES
   ```

   For more information about the installation process, see Installing DataPlane Platform.
4. Bring up the DataPlane Platform service.

   ```
   /usr/dp/1.2.x.x-XX/core/bin/dpdeploy.sh init --all
   ```

   Note: XX indicates the release build number.
5. Run the Docker Container

   ```
   docker ps | awk '{print $NF}'
   NAMES
   dp-app
   dp-cluster-service
   dp-db-service
   ```

```
dp-gateway
knox
dp-consul-server
dp-database
```

Note: Verify if the SELinux module is enabled by repeating step 2.

Make sure that DataPlane Platform UI is running and SELinux module is enabled as well.

Verify the same using https://<dataplane-host>

(Optional) Installing DataPlane Platform using the Non-Root option

You can install DataPlane Platform using the non-traditional procedure.

Note that this procedure is just optional and you continue to use the normal installation procedure. For more information, see Install DP Platform.

You can use the non-root or non-sudo option to install the DataPlane Platform. There are three phases which you must complete and verify that your installation works fine.

- Installing the DataPlane Platform package
- Verifying the DataPlane Administrator commands
- Verifying the Docker Administrator commands

Creating the non-root user

You must create the non-root user in the Docker group.

Procedure

1. Create a Docker group, if it does not exist:

```
root@username-docker ~ $ groupadd docker
```

2. Add the user to the Docker group:

```
root@username-docker bin]# useradd -G docker username
```

Installing DataPlane Platform as non-root user

Perform these actions as part of the installation tasks.

Procedure

1. Install Docker using the access privileged (root/sudo) user.
2. Install the DataPlane Platform using access privileged (root/sudo) user.

   yum install dp-core
3. Change dp directory permissions to 757 (Others permission should be 7).

```
chmod -R 757 /usr/dp
```

4. Switch the user to the username

```
root@username-docker usr]# su username
```

5. Initialize DP using a username, which is part of the Docker group.

   ```
   ./dpdeploy.sh init --all
   ```

6. Run docker ps --format "table {{.ID}}\t{{.Names}}

   ```
   CONTAINER ID        NAMES
   2e6342cb5172        dp-app
   f6c1d76b9c0c        dp-cluster-service
   88a321f0e44e        dp-db-service
   05de38bb55ff        dp-gateway
   a1e589b9d35b        knox
   713f88aa2b3a        dp-consul-server
   c728c2249472        dp-database
   ```

   Other DataPlane Platform administrative tasks can be performed using user (from the Docker group).

New user administration tasks

The newly created user must perform the administative actions.

Procedure

1. Run the process command using dpdeploy.sh

   ```
   username@username-docker bin]$ ./dpdeploy.sh ps
   ```

   ```
   CONTAINER ID       IMAGE        COMMAND        CREATED        STATUS
    PORTS        NAMES

   4dffa7c91e2e  hortonworks/dp-app:1.x.x.x-XX "/bootstrap.sh" 30 minutes ago
    Up 30 minutes 0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp, 9000/tcp dp-app
   22a4533e44c0  hortonworks/dp-cluster-service:1.x.x.x-XX "./
   docker_service_st…" 30 minutes ago Up 30 minutes 9009-9010/tcp dp-cluster-
   service
   59adc48ba163  hortonworks/dp-db-service:1.x.x.x-XX "./docker_service_st…"
    30 minutes ago Up 30 minutes 9000/tcp dp-db-service
   92c251ff37c4  hortonworks/dp-gateway:1.x.x.x-XX "./docker_service_st… 30
    minutes ago Up 30 minutes 8762/tcp dp-gateway
   71e3bac176d3  hortonworks/dp-knox:1.x.x.x-XX "/usr/dp-scripts/kno…" 30
    minutes ago Up 30 minutes 53/udp, 8300-8302/tcp, 8400/tcp, 8301-8302/udp,
    8500/tcp knox
   9e0514521d67  consul:1.0.1 "/bin/sh -c 'mkdir -…" 30 minutes ago Up 30
    minutes 8300-8302/tcp, 8301-8302/udp, 8600/tcp, 8600/udp, 0.0.0.0:8500-
   >8500/tcp dp-consul-server
   a283f8c752ff  postgres:9.6.3-alpine "docker-entrypoint.s…" 31 minutes ago
    Up 30 minutes 5432/tcp dp-database
   ```

2. Verify the metrics for DataPlane

   ```
   username@username-docker bin]$ ./dpdeploy.sh metrics
   ```

   ```
   {"service_metrics":[{"service":"core","instance":"xxx.xx.x.:xxxxx",
   "metrics_url":"http://xxx.xx.x.:xxxxx/metrics"},
   {"service":"cluster-service","instance":"xxx.xx.x.:xxxxx",
   "metrics_url":"http://xxx.xx.x.:xxxxx/metrics"},
   {"service":"db-service","instance":"xxx.xx.x.:xxxxx",
   "metrics_url":"http://xxx.xx.x.:xxxxx/metrics"}],
   "gateway_metrics":{"mem":812491,"mem.free":598894,"processors":4,
   "instance.uptime":753240,"uptime":777257,"systemload.average":0.07,
   "heap.committed":731136,"heap.init":126976,"heap.used":132241,
   "heap":1780736,"nonheap.committed":82944,"nonheap.init":2496,
   ```

```
"nonheap.used":81356,"nonheap":0,"threads.peak":37,"threads.daemon":34,
"threads.totalStarted":44,"threads":37,"classes":10376,"classes.loaded":10376,
"classes.unloaded":0,"gc.ps_scavenge.count":13,"gc.ps_scavenge.time":720,
"gc.ps_marksweep.count":2,"gc.ps_marksweep.time":335,
"gauge.response.service.core.star-
star":302.0,"counter.status.200.service.metrics":1,
"gauge.response.service.db.star-star":478.0,"gauge.response.service.metrics":80.0,
"counter.status.200.service.db.star-star":1,"gauge.response.health":11.0,
"counter.status.200.health":74,"counter.status.200.service.core.star-star":7,
"httpsessions.max":-1,"httpsessions.active":0}}
```

3. Add hosts for DataPlane containers

```
username@username-docker bin]$ ./dpdeploy.sh utils add-host <IP Address>
 <host-name>
```

```
Successfully appended to '/etc/hosts' in dp-app.
Successfully appended to '/etc/hosts' in dp-db-service.
Successfully appended to '/etc/hosts' in dp-cluster-service.
Successfully appended to '/etc/hosts' in knox.

[username@username-docker bin]$ docker exec -it dp-app bash
root@5132e1d78db9:/# cat /etc/hosts
<IP Address> <host-name>

[username@username-docker bin]$ docker exec -it dp-db-service bash
root@1f24870f344f:/# cat /etc/hosts
<IP Address> <host-name>
```

4. Stop Dataplane containers

```
username@username-docker bin]$ ./dpdeploy.sh stop
```

```
Warning!
This command will stop all the DataPlane containers.

Do you want to continue? (yes/no): yes
Destroying containers
dp-database
dp-app
dp-db-service
dp-cluster-service
dp-gateway
knox
dp-consul-server
Stop complete.
```

a) username@username-docker bin]$ docker ps

```
CONTAINER ID    IMAGE   COMMAND   CREATED   STATUS   PORTS     NAMES
[username@username-docker bin]$
```

b) Start DataPlane containers

```
username@username-docker bin]$ ./dpdeploy.sh start
```

Enter previously entered master password for DataPlane Service: <Maser password>.

```
Using previously generated self-signed certificates (for example purpose
 only)
a2c5a8447e69761490ea620d4bc016a8657919c64d1b5d0604a9ad6ec9ace778
Initializing app
```

```
Starting Consul
576e046cf653f091217b3689dd44d69ad29df12b21fcb3d6f78f5421a0b4dd6b
Starting Knox
3e9167fe5968e57c27c349442dde7935714b8fd1d8212853e694b2b54f17c653
Starting Database (Postgres)
Starting Gateway
e99511703a94d46693a48458cdcf74fe44c2f659c469986c36a6715717caa74d
Starting DB Service
78bb22e0ab10470c73fe37cfa784630977782ef161a55e0741127414c83af646
Starting Cluster Service
ae5d4f6c9b988a5bbc853e822a4ce78191cea244ba409a16177a07aadef8e67c
Starting Application API
86f82b8b76b2f1fa6f13f8ce7760381162ee767df5bc3b084be2853281a3d63e
Initialization and start complete.
```

5. Verify DataPlane container logs

```
username@username-docker bin]$ ./dpdeploy.sh logs -f dp-app
```

```
Reloading nginx configuration: nginx.
06:16:35 INFO  Slf4jLogger - Slf4jLogger started
06:16:38 INFO  PluginManifestService - reading /plugins/cloudbreak.json
06:16:39 INFO  PluginManifestService - reading /plugins/das.json
06:16:39 INFO  PluginManifestService - reading /plugins/dlm.json
06:16:39 INFO  PluginManifestService - reading /plugins/dss.json
06:16:39 INFO  PluginManifestService - reading /plugins/smm.json
06:16:40 INFO  Play - Application started (Prod)
06:16:40 INFO  NettyServer - Listening for HTTP on /0.0.0.0:xxxxx
06:16:42 INFO  application - Registered service
 DpService{serviceId='core_xxx.xx.x.x:xxxxx', serviceTags=[core-service],
 port=xxx.xx.x.x, port=xxxxx}

06:16:43 WARN  application - Gateway discovery failed, endpoints
 configured in config will be used
Reloading nginx configuration: nginx06:16:47 INFO  application - Running a
 service check for serviceId core_xxx.xx.x.x:xxxxx
.
06:16:52 INFO  application - Running a service check for serviceId
 core_xxx.xx.x.x:xxxxx
Reloading nginx configuration: nginx.
06:16:57 INFO  application - Running a service check for serviceId
 core_xxx.xx.x.x:xxxxx
06:17:02 INFO  application - Running a service check for serviceId
 core_xxx.xx.x.x:xxxxx
```

Performing Docker administration tasks

The newly created user must perform the Docker administative actions.

Procedure

1. Run the process command:

```
[username@username-docker bin]$ docker ps --format table
 {{.ID}}\t{{.Names}}
```

```


CONTAINER ID        NAMES
caf97ed61576        dp-app
ba94e608c099        dp-cluster-service
4081ec53d41e        dp-db-service
```

```
ad5344a9a0a1        dp-gateway
01cf21a2ff8a        knox
df08ef327507        dp-consul-server
6378a8022f12        dp-database
```

2. The newly created user must verify the Docker log files. Run the command:

```
[username@username-docker bin]$ docker logs -f dp-app
```

```
Reloading nginx configuration: nginx.

06:16:35 INFO  Slf4jLogger - Slf4jLogger started
06:16:38 INFO  PluginManifestService - reading /plugins/cloudbreak.json
06:16:39 INFO  PluginManifestService - reading /plugins/das.json
06:16:39 INFO  PluginManifestService - reading /plugins/dlm.json
06:16:39 INFO  PluginManifestService - reading /plugins/dss.json
06:16:39 INFO  PluginManifestService - reading /plugins/smm.json
06:16:40 INFO  Play - Application started (Prod)
06:16:40 INFO  NettyServer - Listening for HTTP on /0.0.0.0:xxxxx
06:16:42 INFO  application - Registered service
 DpService{serviceId='core_xxx.xx.0.x:xxxxx', serviceTags=[core-service],
 port=xxx.xx.0.x, port=xxxxx}
06:16:43 WARN  application - Gateway discovery failed, endpoints
 configured in config will be used
Reloading nginx configuration: nginx06:16:47 INFO  application - Running a
 service check for serviceId core_xxx.xx.0.x:xxxxx
.
06:16:52 INFO  application - Running a service check for serviceId
 core_xxx.xx.0.x:xxxxx
Reloading nginx configuration: nginx.
06:16:57 INFO  application - Running a service check for serviceId
 core_xxx.xx.0.x:xxxxx
06:17:02 INFO  application - Running a service check for serviceId
 core_xxx.xx.0.x:xxxxx
```

Copying the Docker CP from local directory

You must verify if the new user can copy the Docker CP from you local diretory to the Container.

About this task

Run the following commands.

Procedure

1. [username@username-docker bin]$ ls /usr/dp/1.x.x.x-XX/bin/dpdeploy.sh

   dpdeploy.sh
2. [username@username-docker bin]$ docker cp dpdeploy.sh dp-app:/tmp/
3. [username@username-docker bin]$ docker exec -i dp-app ls /tmp

   dpdeploy.sh

Copying the Docker CP from the Container

You must verify if the new user can copy the Docker CP from the Container to your local directory.

About this task

Run the following commands.

Procedure

1. [username@username-docker bin]$ docker cp dp-app:/tmp/dpdeploy.sh /tmp
2. [username@username-docker bin]$ ls /tmp/dpdeploy.sh

   /tmp/dpdeploy.sh

Supported databases

DataPlane platform supported the following databases:

- PostgreSQL
- MySQL

Note: As per the DataPlane Platform standards, you must use the backend database name as dataplane only.

Configure PostgreSQL external database

DataPlane includes PostgreSQL database. It is available as embedded database which is intended for non-production use. It is strongly recommended to use an external database for production environments. After installing the database following the instructions provided with the database software, you must set up the database for use with DataPlane.

Before you begin

- PostgreSQL database is supported in this version. PostgreSQL can be used in two ways - Embeded Database - For non production/evaluation purposes and External Database - For Production environments.
- Be sure to have the database URI, username, and password available.
- The PostgreSQL database server must have been installed and properly configured for remote access.
- A database must have been created.
- A database user must have been created and assigned permissions for the new database.

Procedure

1. Open the <installer_home>/config.env.sh file for editing.

   ```
   vi /usr/dp/current/core/bin/config.env.sh
   ```

2. Modify the DB Configs settings to add the appropriate connection information.

   ```
   USE_EXTERNAL_DB="yes"
   DATABASE_URI="jdbc:postgresql://<host_name>:5432/dataplane"
   DATABASE_USER="<user_name>"
   DATABASE_PASS="<password>"
   ```

   If you need to connect to the database through an SSL connection, modify the DATABASE_URI parameter in the above example as follows:

   ```
   DATABASE_URI="jdbc:postgresql://<host_name>:5432/dataplane?
   ssl=true&sslfactory=org.postgresql.ssl.NonValidatingFactory"
   ```

   The PostgreSQL database is configured.

Configure MySQL external database

It is strongly recommended to use an external database for production environments. After installing the database following the instructions provided with the database software, you must set up the database for use with DataPlane.

Before you begin

- The MySQL database server must have been installed and properly configured for remote access.
- A database must have been created. Note that you should create the DB which is UTF-8 encoding compliant.
- You can use the following queries to configure database for DataPlane. Make sure to replace username and password.

```
CREATE DATABASE dataplane CHARACTER SET utf8 COLLATE utf8_general_ci;
CREATE USER username IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON dataplane.* TO username@'%';
flush privileges;
```

- Download the MySQL JDBC driver.

  You must download MySQL driver from the following URL:

  MySQL Driver Download.

Procedure

1. Open the <installer_home>/config.env.sh file for editing.

   ```
   vi /usr/dp/current/core/bin/config.env.sh
   ```

2. Modify the configuration parameters as shown below.

   ```
   USE_EXTERNAL_DB="yes"
   DB_TYPE="MySQL"
   DATABASE_URI="jdbc:mysql://<host_name>:<port>/dataplane"
   DATABASE_USER="<username>"
   DATABASE_PASS="<password>"
   MYSQL_DRIVER_LOCATION="</path/to/mysql/mysql-connector-java-8.0.19.jar>"
   ```

   Note: You must update the parameter values as per your production environment. Make sure to update **MySQL_DRIVER_LOCATION** variable with MySQL driver JAR file (absolute path) as well.

   If you need to connect to the database through an SSL connection, modify the DATABASE_URI parameter in the above example as follows:

   ```
   DATABASE_URI="jdbc:mysql://<host_name>:<port>/dataplane?
   verifyServerCertificate=false&useSSL=true"
   ```

   The MySQL database is configured.

(Optional) Change the default DP ports

During DataPlane installation, the installer checks if the default DP ports are available. If they are not available, you must change the default DP ports.

About this task

Change the default DP ports.

Procedure

1. Open the configuration file:

   ```
   /usr/dp/current/core/bin/config.env.sh
   ```

2. Modify the details of the ports in the configuration file.

   ```
   APP_HTTP_PORT="<port-number>"
   APP_HTTPS_PORT="<port-number>"
   ```

(Optional) Configure a TLS certificate

If you choose to use the default TLS (formerly SSL) certificates provided with DataPlane, then DataPlane generates self-signed certificates. If using your own certificates, then you must modify certificate configuration settings.

About this task

For private keys, we currently support only encrypted keys generated with 3DES algorithm. This is the default algorithm used by OpenSSL v1.1.x.

Before you begin

- You will be configuring your own SSL certificate.
- Have the full path and file name for the public key and private key (.pem files) and the certificate password available.

Procedure

1. Open the configuration file:

   ```
   /usr/dp/current/core/bin/config.env.sh
   ```

2. Uncomment and modify the following properties:

   ```
   USE_TLS="true"
   USE_PROVIDED_CERTIFICATES="yes"
   DATAPLANE_CERTIFICATE_PUBLIC_KEY_PATH="<absolute_path_of_public_key_file>"
   DATAPLANE_CERTIFICATE_PRIVATE_KEY_PATH="<absolute_path_of_encrypted_private_key_file>
   ```

3. Save the file.

Initialize DP Platform

After installing the RPMs and optionally configuring your external database and TLS certificate, you must initialize DataPlane.

About this task

You will be initializing and configuring the DP Platform.

Before you begin

If you plan to use an external database or use your own TLS (SSL) certificate, be sure to configure those options prior to initializing the DP Platform. Refer to Configure an external database and Configure a TLS certificate for more details.

Procedure

1. Navigate to the folder containing the DataPlane configuration script.

   ```
   cd /usr/dp/current/core/bin
   ```

2. Run system checks needed by DP

   ```
   ./dpdeploy.sh system-check
   ```

   system-check option allows user to run couple of system checks needed by DPS. When executed, it will check:

   - if the required ports are available
   - if docker client is running and docker daemon is running
   - if SELinux is in permissive mode
   - system firewall settings
   - iptables rules
   - IP Forwarding

3. Initialize the software.

   This loads the Docker images into your local system and prompts for configuration options.

   ```
   ./dpdeploy.sh init --all
   ```

   You can use disable-system-check flag to skip system checks although it is not a recommended way of installation.

   ```
   ./dpdeploy.sh init --all --disable-system-check
   ```

4. Create the password for a Super User and a Master Password for the system.

   Ensure that you remember these passwords, as they cannot be retrieved or reset. If you forget the password, contact Hortonworks Support for further assistance.

   ```
   Setup the default 'admin' user. This user is for initial setup of DataPlane,
    including configuration of LDAP and adding additional DPS Admins.

   Enter a password for this 'admin' 'user:
   Re-enter password:

   Setup a Master password for the DataPlane. The Master password is used to
    secure the keystore storage for the system.
   Therefore, it is recommended you use a strong password. You will need
    to provide the Master password for various DataPlane administrative
    operations.

   Important: The Master password cannot be reset easily. Do not forget or
    misplace.

   Enter master password for DataPlane Service (Minimum 6 characters long):
   Reenter password:
   ```

5. When the initialization process completes, you can check the status of the docker containers using the following command:

   ```
   docker ps
   ```

   Sample output:

   ```
   IMAGE                                       COMMAND                  CREATED
                STATUS                    PORTS
                                        NAMES
   ```

```
7054de3e2e78        hortonworks/dp-app:1.2.1.0-37               "/
bootstrap.sh"          2 minutes ago       Up 2 minutes        0.0.0.0:80-
>80/tcp, 0.0.0.0:443->443/tcp, 9000/tcp                          dp-app
c7119441ef77        hortonworks/dp-cluster-service:1.2.1.0-37   "./
docker_service_st…"   2 minutes ago       Up 2 minutes        9009-9010/tcp
                                                              dp-cluster-
service
7651d09d33d9        hortonworks/dp-db-service:1.2.1.0-37        "./
docker_service_st…"   2 minutes ago       Up 2 minutes        9000/tcp
                                                             dp-db-service
e3bc2ac15dc1        hortonworks/dp-gateway:1.2.1.0-37           "./
docker_service_st…"   2 minutes ago       Up 2 minutes        8762/tcp
                                                             dp-gateway
98bf7858cb13        hortonworks/dp-knox:1.2.1.0-37              "/usr/dp-
scripts/kno…"   2 minutes ago       Up 2 minutes        53/udp, 8300-8302/
tcp, 8400/tcp, 8301-8302/udp, 8500/tcp                         knox
05a79ed55f05        consul:1.0.1                                "/bin/sh
 -c 'mkdir -…"   2 minutes ago       Up 2 minutes        8300-8302/tcp,
 8301-8302/udp, 8600/tcp, 8600/udp, 0.0.0.0:8500->8500/tcp   dp-consul-
server
d0cd58aaef00        postgres:9.6.3-alpine                       "docker-
entrypoint.s…"   2 minutes ago       Up 2 minutes        5432/tcp
                                                        dp-database
```

Output descriptions:

Docker Container Name	Description
dp-app	DP Instance application (UI, web, etc.), accessible from port 443 (port 80 redirects to port 443) by def
dp-cluster-service	Powers how DP Platform talks to clusters
dp-db-service	Backend data store API in support of DP Apps
dp-gateway	Handles routing between DP Platform, DP Apps, Knox, etc.
knox	Runs a Knox instance that wraps the DataPlane instance. This is the AuthN enforcement point for Dat (SSO).
dp-consul-server	Handles the networking of the containers
dp-database	The PostgreSQL database instance that the DP instance uses by default. This is not used if you configu external database.

6. Browse to your DP Instance host and proceed to log in using the Super User admin account.

   Note:

   If you are using AWS, do not use the public DNS to access DataPlane. Use a public IP address or set up and use a DNS (Route 53) fully qualified domain name (FQDN).

   ```
   https://<DP_host_FQDN>
   ```

   As part of the installation process, data collection using cookies and other telemetry mechanisms is turned on by default. To disable all data telemetry, see the *DataPlane Administration Guide* for disabling data telemetry.

   Note: If you get an error message when configuring the platform, destroy the setup by running the ./dpdeploy.sh destroy --all command and restart the initialization process.

What to do next

You can now complete configuration of the DP Platform.

Related Tasks

(Optional) Configure a TLS certificate

Log in and configure DP Platform

To complete your installation, you must log in and configure the DP Instance for LDAP.

About this task

You must configure your DP Instance for LDAP and set up your initial DataPlane Admin users and groups.

Before you begin

Be sure you have your enterprise LDAP configuration available. Refer to *Enterprise LDAP requirements* for more information.

Procedure

1. Browse to your DP Instance host.

   Use a public IP address or set up and use a DNS (Route 53) fully qualified domain name (FQDN).

   https://<DataPlane-host-FQDN>
2. Log in using the Super User admin account you configured during initialization:

   Username: admin
   Password: Use the password specified during DataPlane initialization

   After login, the initial “Onboarding Welcome” screen displays.
3. Click **Get Started** to proceed with setting up authentication.

   The Onboard/Configure LDAP page displays the Setup Authentication settings.
4. Enter your LDAP information.

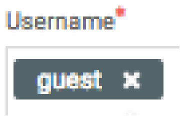
   See Enterprise LDAP requirements for more details on these settings and options.

   If you are using LDAPS and a self-signed certificate, be sure to click *Upload certificate* and provide your certificate in order for DataPlane to connect to your LDAP instance. Use the corporate LDAP values that you collected during the preparatory steps.
5. Click **Save**.
   A success message displays on the page.
6. Add users and groups, which will be the initial members of the default DataPlane Admin role.

   At least one user must be configured and must have at least the DataPlane Admin role, which is needed to add other users or groups, assign them roles, and configure services in DataPlane.

   Tip:

   You must click the name of the user when it displays and ensure it appears in the Users field on a dark background.

   Username*
   guest ✖

   If the name appears on a white background, it means the name is not recognized and the action fails.
7. Click **Save & Login** to save your changes.
8. Log in as one of the DataPlane Admin users you added using the LDAP credentials of the Admin.

   Note: In case the login fails, it is most likely due to an error in the LDAP setup. Verify the settings with your LDAP admin and then log in using the Super admin user using the https://dataplane-host/sign-in page. After logging in, navigate to the User Management page. Click Edit LDAP settings from the righthand drop down menu on the LDAP information section. Edit the properties that should be corrected and save.

Results

You are now ready to proceed to managing your DP platform, including cluster registration and DP App installs.

What to do next

You can now install and configure additional DP Apps and manage your DP Instance.

Related reference

Enterprise LDAP requirements

Configure Knox and Ranger for registering clusters in DataPlane

Before proceeding further for using DataPlane, you must configure either Knox Trusted Proxy Pattern or Knox SSO and Knox Gateway for DataPlane. It is also highly recommended to configure Ranger to restrict access to DataPlane.

Plan for Trusted Proxy Pattern Configuration

The communication between DataPlane Platform and cluster services such as Data LifeCycle Manager requires Knox. You can choose to configure Knox Trusted Proxy Pattern or Knox SSO.

DP Platform and the DP Apps leverage Knox to provide users and services with simplified and consistent access to clusters, data, and other services. DataPlane authenticates users against a centralized identity provider in the organization (such as an LDAP or AD). Having Knox set up with your clusters ensures that those users and services are authorized to perform specific actions on the respective clusters, and propagates the identity of the user or service from DataPlane to the cluster services.

Configuring and using Knox SSO involves manual setup of Knox topologies and manual registration of Data Plane services. To configure Knox SSO to use DP apps such as DLM and DSS, you must manually set up the Knox topologies such as token.xml and dpproxy.xml. In addition, to log in to services such as Ambari or DP apps such as DLM, you need to log in through a Knox SSO page.

Trusted proxy pattern provides an alternative secure way for DataPlane to communicate with Ambari. It eliminates the need to enable Single Sign On on the services.

Knox Trusted Proxy Pattern addresses this problem of varying authentication screens as the clusters are already registered using the DP Cluster Setup Utility Script.

The DP Cluster Setup Utility Script performs the following functions:

- Validates if the environment meets all prerequisite requirements
- Checks if all the DP App agents such as DLM Engine and DP Profiler Agent are installed
- Adds and configures the Knox topologies that DP requires and the installed apps require to communicate with cluster services - token.xml and dp-proxy.xml
- Sets up the trusted proxy configuration for services
- Registers the clusters in DataPlane

If your environment meets the requirements for Knox Trusted Proxy Pattern, it is recommended that you configure Knox TPP instead of Knox SSO. See Prerequisites for Knox Trusted Proxy Pattern for more information about these requirements.

Prerequisites for Knox Trusted Proxy Pattern

To configure Knox TPP in your cluster to work with your DP instance, make sure the required prerequisites are met.

- Make sure that the cluster is running Ambari 2.7.3 or later version.
- Make sure that Kerberos is enabled on the cluster.
- Make sure that Ambari is configured for Kerberos.
- Make sure that Knox is installed on the cluster.
- Make sure that Ranger is installed on the cluster.
- Make sure that the cluster has your DP App cluster agents installed and any dependent cluster services installed and configured.
- You will run this command from your cluster Knox gateway host as root.
- Make sure you have the following information readily available:
    - Ambari URL (with an Ambari Admin username and password)
    - DataPlane URL (with a DataPlane Admin username and password)

Configure Knox Trusted Proxy Pattern for DataPlane

This topic provides an overview of how to configure Knox Trusted Proxy Pattern (TPP) in your cluster to work with DataPlane.

About this task

You will be configuring Knox TPP in your cluster to work with your DP instance.

Procedure

1. In a terminal, SSH to the Knox host in your cluster.
2. Download the DP Cluster Setup Utility Script from this location.

   ```
   export RELEASE=<release-name>;curl https://raw.githubusercontent.com/
   hortonworks/dp-cluster-reg/master/install.sh |sh
   ```

   Make sure to replace <release-name< with 1.2.2.
3. Execute the script as root user and follow the prompts.

   ```
   python dp-cluster-setup-utility.py
   ```

   Note: Knox Trusted Proxy Pattern is not supported in HA mode. Make sure to manually copy all the setup files to Knox instances.

Configure Knox SSO for DataPlane

This topic provides an overview of how to configure Knox SSO in your cluster to work with DataPlane. Refer to the Hortonworks Data Platform or Hortonworks DataFlow documentation for details that might be applicable to your specific cluster configuration and setup.

About this task

Note: As part of configuring Knox SSO to work with DataPlane, you will be setting up Knox topologies for token exchanges to allow your DP instance to communicate and handle SSO requests. It is strongly recommended that in your cluster, you configure Ranger to restrict access to these token topologies to be only from your DP instance. See Configure Ranger in your Cluster section for more information.

- You will be configuring Knox SSO in your cluster to work with your DP instance.

Before you begin

- You must have installed and configured DataPlane.
- Minimally, Knox SSO should be configured for Ambari.

  Note: If you are using Ambari 2.7 or later, Ambari provides a helper "setup-sso" command to simplify the setup of Knox SSO for Ambari and certain cluster services. Refer to the *Ambari Security Guide* for more information.
- Knox host FQDN must be DNS addressable and available from your DataPlane environment. If your Knox configuration is setup for High Availability (HA) with more than one Knox instance running behind a proxy, the FQDN/IP of that proxy must be DNS addressable and available from your DataPlane environment.

  If it is not, the Knox IP address must be in the /etc/hosts file on the DP environment. Refer to the *DP Administration Guide*for details on how to add Knox to the DataPlane environment hosts.
- You must have an SSL certificate (such as a .pem file) available and have access to the public key in the file.

Note: Use the following information to register the cluster in DataPlane. Currently, DataPlane does not allow changing the value of cookie to anything other than hadoop-jwt. This value is used internally by DataPlane.

<param><name>knox.token.client.data</name>

<value>cookie.name=hadoop-jwt</value></param>

Procedure

1. In a terminal, SSH to the DP host.
2. Navigate to $DP_INSTALL_HOME/certs/.

   ```
   cd /usr/dp/current/core/bin/certs/
   ```
3. Display the content of the ssl-cert.pem file.

   ```
   cat ssl-cert.pem
   ```
4. Copy and retain the DataPlane public key displayed in the certificate between "Begin Certificate" and "End Certificate", because you need it in a succeeding step.

   The public key looks similar to the following:

   -----BEGIN CERTIFICATE-----

```
MIIEpDCCAowCCQDgyOmhg8r5wjANBgkqhkiG9w0BAQsFADAUMRIwEAYDVQQDDAlk
YXRhcGxhbmUwHhcVEBgwNjA2MDMzMzEyWhcNMTkwNjA2MDMzMzEyWjAUMRIwEAYD
VQQDDAlkYXRhcGxhbmUwggIiMYu7ncDA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAo
fTf6/5drxlYa5EeHDetQo3I50Vx+Tj9jpd8t1x+3zJMO3xI6UCtHFiZlxS8IToTw
V2BEOPH1K7qqRQjTagZtqNU7JNiEouBxO+lRYXdyaqhCIIUcgspsop9W5C9T2aM6
HWx73MapoP5r4dRTpYCITWJW7GkvJGhplIK51MhWD2dL9+bzsZ/sY9nwGJ2iUZ39
VGKMRC8TQlR2uwmRl2GziOCjMrfJIDDBxNm2xCbpYbNo8prle2EcntLgrw/EBNyA
Fbp8a506gL7TbbjnqFM4iHhr7vVzbOzl14Qci+GLdTl51yNLER1k5sC/TEFpKBXP
IcY8UOuU6bbXrF/ZYR5FjYu7ncDOp0wDLz0lmLPwU27tE9D9SR+k0PYo8xvLIw31
iPsw0UuF4ouWJI9UaZ2i0vGJJcU8TtH7cOEyUL+wv0sCNHp5eI1hXgdCTkyFZv7s
2xWNA12L0TSd/49nHA9fIrXHYp5inmHrJiRP4wJlxFDvbZFltFepaG4I2pWhdrwY
2Sj0G084J294FNLad2xpoIacwaKWzdJ6HTvYo09Pqiu0YAchGicGCjwUU0omeK61
mZRmwENGYlzKXsrBAAGU1QuzZTFevA36d71iCcTZOwKp7KeHL0e8RJ7fTfgEXvRT
WzmnVdOCfpflmaXoVhyATIRVAF5RDdnYdK+QUPDpVwIDAQABMA0GCSqGSIb3DQEB
ViNN1eA4ICAQCNpcF0UJVAs7OhAf3DE2IjdWoiiUUr1x8jt5icWOS48Mw8mM9ADd
tGZ3IMrSB3qerixBN7tlFS/NCTXvS78imNXJT5xnJvNUYekWj/uLi4Vh9/Vg/l/2
zm7uLn5gTLwzDr9QGUF8Hb7+o/nW3pYLwrSoay5ApIysMd/mOykzFuLf4P7CHOMe
cFImfN82xSD46W2zLbW/aJ5jdwXRa27L075TA91sqRRO0tObn+vCBgsehvT1EBFy
```

```
U7puKS9fv2QHMS7rmt3ixhWw5AJ8wG9D4/nJq0cJhIKaqiDn9nqGiY8GBUwa/YAc
tAuIqo+LcvoIr/J+yc+7SXxDJXM+SlbS+lA0Fp/EdIMuDbey2T6Sabor4khzi78E
PeCBKHnZ2V8MTtcAXKw6RTSToBhIGozm9oRGU1xfT61xAein+ba4UH1yZUanna7o
NbRQIKjSYC48oxYyWEA6HlZbTjy/uaBU3WP78mcoYUfFronK7fGkGvB8+xMYlYc+
aM1t87Z/KpY2d2CtDEG6qMl5wWTJqwocN5cYNwubgJM8vt1uD1IhsezHjr1Tuv1x
988ztA/raNLF921ZAc5W1Xly5JF4z1lhZQZqDBpMsdo05HWKU4bgdLLoCN7bFVPX
Sclm0TtyUSXblXfPqXqBfnPJNiimmLk1+SmOxX9h+dOHfcNMsSNa5g==
```

-----END CERTIFICATE-----

5. On your cluster Knox host, create three topology files - token.xml, redirect.xml, and redirecttoken.xml topology files.

   ```
   vi /etc/knox/conf/topologies/token.xml
   ```

   ```
   vi /etc/knox/conf/topologies/redirect.xml
   ```

   ```
   vi /etc/knox/conf/topologies/redirecttoken.xml
   ```

   Note: The redirecttoken.xml topology file should be exactly same as the token.xml topology file. For security purposes, the TTL of the token should be kept very low. It is recommended to keep the value at 10 seconds.

6. Add the required content to the token.xml file on each cluster host running a Knox instance:

   a) Add the basic topology content.

      You can copy and paste the following content into the file and modify the content as needed.

      ```
      <?xml version="1.0" encoding="UTF-8"?>
      <topology>
         <uri>https://$KNOX_HOSTNAME_FQDN:8443/gateway/token</uri>
         <name>token</name>
         <gateway>
            <provider>
               <role>federation</role>
               <name>SSOCookieProvider</name>
               <enabled>true</enabled>
               <param>
                  <name>sso.authentication.provider.url</name>
                  <value>https://$KNOX_HOSTNAME_FQDN:8443/gateway/knoxsso/api/
      v1/websso</value>
               </param>
               <param>
                  <name>sso.token.verification.pem</name>
                  <value>
                      $ADD_THE_PUBLIC_KEY_HERE
                  </value>
               </param>
            </provider>
            <provider>
               <role>identity-assertion</role>
               <name>HadoopGroupProvider</name>
               <enabled>true</enabled>
            </provider>

         </gateway>

         <service>
            <role>KNOXTOKEN</role>
            <param>
               <name>knox.token.ttl</name>
      ```

```
         <value>100000</value>
      </param>
      <param>
         <name>knox.token.client.data</name>
         <value>cookie.name=hadoop-jwt</value>
      </param>
      <param>
         <name>main.ldapRealm.authorizationEnabled</name>
        <value>true</value>
      </param>
   </service>
</topology>
```

Provide the following details in the topology file:

Property	Values	Description
sso.token.verification.pem	Certificate	Paste in the public key value that you copied in a previous step, replacing $ADD_THE_PUBLIC_KEY_HERE (be sure to exclude the BEGIN CERTIFICATE and END CERTIFICATE text).
knox.token.ttl	milliseconds	Expiry time of the token. A value of -1 means no expiry. For security purposes, the TTL of the token should be kept very low. It is recommended to keep the value at 10 seconds (10000).
sso.authentication.provider.url	Knox SSO URL	The URL to your cluster Knox SSO endpoint. Replace $KNOX_HOSTNAME_FQDN with the fully qualified domain name of the host.
identity-assertion	true | false	Enables the “HadoopGroupProvider” Hadoop user-group mapping, which identifies the groups to which users belong

Note: The authorization=XASecurePDPKnox parameter and main.ldapRealm.authorizationEnabled=true parameter enable Ranger authorization with the token topologies in Knox.

7. Add the required content to the redirect.xml file on each cluster host running a Knox instance:
   a) Add the basic topology content.
   b) You can copy and paste the following content into the files and modify the content as needed.

```
<topology>
    <name>tokensso</name>
    <gateway>
        <provider>
            <role>federation</role>
            <name>JWTProvider</name>
            <enabled>true</enabled>
        </provider>
        <provider>
            <role>identity-assertion</role>
            <name>Default</name>
            <enabled>true</enabled>
        </provider>
    </gateway>
    <service>
        <role>KNOXSSO</role>
        <param>
            <name>knoxsso.cookie.secure.only</name>
            <value>true</value>
        </param>
        <param>
            <name>knoxsso.token.ttl</name>
            <value>600000</value>
        </param>
        <param>
            <name>knoxsso.redirect.whitelist.regex</name>
            <value>^https?:\/\/(DOMAIN_OF_CLUSTER|localhost|127\.0\.0\.1|
0:0:0:0:0:0:0:1|::1):[0-9].*$</value>
        </param>
    </service>
</topology>
```

Provide the following details in the topology file:

Property	Values	Description
knoxsso.cookie.secure	true | false	This sets if a session cookie is require or not. If your cluster Ambari is configured for SSL, then set this value to true. Otherwise, set to false.This value is true if the secure cookie is required.
knox.token.ttl	milliseconds	Expiry time of the token. A value of -1 means no expiry.
knoxsso.redirect.whitelist.regex	regex	This should have the regex which matches the URL in the to or original Url query parameter for the two separate calls. Be sure to replace “DOMAIN_OF_CLUSTER” in the example regex provided.

8. Verify that Knox has picked up the files:

a) Log in to the Knox-enabled node.

b) Ensure that a directory called token.topo.<number> is present in the path /var/lib/knox/data-<version>/deployments/.

If the files are not present, verify that the content in the token.xml file is correct. You can check the Knox gateway logs for error information.

Related Concepts

Knox SSO with DataPlane clusters

Related Tasks

Advanced: Add host entries to the DataPlane environment

Related Information

Defining Cluster Topologies

Configure Knox Gateway for DataPlane

DP Platform communicates with services on the cluster like DP Agents, Ambari, Atlas, Ranger, etc as well as DP Agents used by DP Apps (for example: DLM Engine for DLM and Profiler for DSS). To eliminate DataPlane communicating directly to all the cluster service endpoints, you can configure Knox Gateway as a proxy to your cluster services.

About this task

⚠ **Important:** If you are using TLS wire encryption on your clusters, you must configure Knox Gateway to proxy requests to and from DP host.

This topic provides an overview of how to configure Knox Gateway proxy in your cluster services for DataPlane communication. If you configure Knox Gateway as the proxy for communication, be sure all DP services are configured through the gateway. Refer to the Hortonworks Data Platform or Hortonworks DataFlow documentation for details that might be applicable to your specific cluster configuration and setup.

Before you begin

Knox host FQDN must be DNS addressable and available from your DataPlane environment. If not, the Knox IP address must be in the /etc/hosts file on the DP environment. Refer to the *DataPlane Administration* guide for details on how to add Knox to the DataPlane environment hosts.

Procedure

1. On your cluster Knox host, navigate to the Knox topologies directory.

```
cd /etc/knox/conf/topologies
```

2. Create a DataPlane proxy topology file.

```
vi dp-proxy.xml
```

3. Add the host name for each of the services listed in the file, based on where that service is running in your cluster.

Tip: At this point, you can add to the file the DP service agents that you plan to install, or you can add them later.

Important:

- Do not modify the URL in the provider section of the file.
- Be sure to keep this file updated if you move services or add services in your cluster.

The <localhost> entry in the following example might be something like ctr-exxxx-xxx-xxx.company.site:20070.

```
Topology dp-proxy.xml

<?xml version="1.0" encoding="utf-8"?>
<topology>
  <gateway>
    <provider>
        <role>federation</role>
        <name>SSOCookieProvider</name>
        <enabled>true</enabled>
        <param>
            <name>sso.authentication.provider.url</name>
            <value>https://localhost:8443/gateway/knoxsso/api/v1/websso</
value>
        </param>
      </provider>
    <provider><role>identity-assertion</role>
      <name>Default</name>
      <enabled>true</enabled>
    </provider>
   </gateway>

  <service>
    <role>AMBARI</role>
    <url>http://<localhost>:8080</url>
  </service>
  <service>
    <role>AMBARIUI</role>
    <url>http://<localhost>:8080</url>
  </service>
  <service>
    <role>RANGER</role>
    <url>http://<localhost>:6080</url>
  </service>
  <service>
    <role>RANGERUI</role>
    <url>http://<localhost>:6080</url>
  </service>
  <service>
    <role>ATLAS</role>
    <url>http://<localhost>:21000</url>
  </service>
  <service>
    <role>ATLAS-API</role>
    <url>http://<localhost>:21000</url>
  </service>
  <service>
   <role>BEACON</role>    ##The DLM Engine
   <url>http://<localhost>:25968</url>
  </service>

  <service>
   <role>PROFILER-AGENT</role>    <!-- The DSS Agent -->
   <url>http://<localhost>:21900</url>
  </service>

</topology>
```

Note: If you plan to set up using Knox gateway, verify your URLs for registration.

(Optional) Configure Ranger to restrict access to DataPlane

It is strongly recommended that in your cluster, you configure Ranger to restrict access to these DataPlane specific topologies to be only from your DP instance, in order to restrict access to only authorized users of DataPlane Platform.

About this task

As part of configuring Knox SSO to work with DataPlane, you setup Knox topologies to allow your DP instance to communicate and handle SSO request token between DP and your cluster.

Note: This is the basic Ranger policy setup to restrict access to the Knox topology to only DataPlane. Additional policies may be recommended or required based on the DP Apps (and their requisite Cluster Agents) you use.

- You will be configuring a Ranger policy to restrict access to Knox SSO token topologies to DataPlane users and your DP Instance.
- You must have installed and configured DataPlane.
- You must have configured Knox SSO for DataPlane. See *Configuring Knox SSO for DataPlane* for more information.
- You must have Ranger installed and configured in your cluster.
- Be sure to also add the authorization role to the token topologies you configured for DP in your Knox SSO setup.

```
<provider>
   <role>authorization</role>
   <name>XASecurePDPKnox</name>
   <enabled>true</enabled>
</provider>
```

Procedure

1. In your cluster, navigate to the Ranger UI and log in.
2. Click **Access Manager**, and then click the Knox repository link, for example: **<cluster-name> Policies**.
3. Click **Add New Policy**, and then enter the following values:

Parameter	**Value**
Policy Type	Access
Knox Topology	token
Knox Service	*

4. Enter groups or user names in **Select Group** or **Select User**.
5. Optional: Under Policy Conditions click **Add Condition** and enter the IP addresses of the DataPlane host.

   This adds an IP-based filter to ensure that only known DataPlane Core hosts can access cluster services through the token topology.
6. Under Permissions, click **Add Permission** and select **Allow**.

Upgrade DataPlane

Make sure you take regular backups of the instance before your proceed with the upgrade procedure.

About this task

Note: To upgrade from one version to another, you must make note of the existing version and the new upgrade version. You must run the upgrade command from the new upgrade version repository folder and enter the folder details of the existing version.

Procedure

1. Back up your existing DataPlane repository (dp.repo) file in the .repo format.
2. Download the upgrade repository tarball to the repository folder:

```
 wget -nv <upgrade-repo-URL> -O /etc/yum.repos.d/dp.repo
```

3. Verify that the repository is downloaded:

```
 yum search dp-core
```

   You should see two dp-core repositories.
4. Update the repository by running the following command:

```
yum update dp-core
```

   You should see two versions of the DP Platforms.
5. Run the upgrade command as follows:

```
./dpdeploy.sh upgrade --from /usr/dp/1.2.0.0-392/core/bin/
```

The following message appears:

```
This will update database schema which cannot be reverted. All backups
 need to be made manually. Please confirm to proceed (yes/no):
```

6. Enter yes to continue.
7. When prompted, enter the master password for DataPlane.

   Note: The master password should be the same as the password used in the previous version of DataPane.

8. If asked, enter the password for supplied certificates. You will prompted only if you have provided certificates in the config.env.sh file.

   A message appears that the upgrade process is complete.

9. Once the upgrade process is successfully completed, add hosts using the following command:

   ```
   ./dpdeploy.sh utils add-host <Host URL> <Host FQDN>
   ```

   Make sure you run this command for each host to add the host entries to all the DP containers.

   Note: This step is required only if you want to add the host entries to the DP environment for the communication to work before the upgrade.

Troubleshooting DataPlane Installation

You can troubleshoot various installation issues such as cluster registration errors, logging issues, and Docker container errors.

Cluster Registration Error Messages

Following are errors you might encounter while registering a cluster in DataPlane on the Add Your Cluster page. Some possible causes and possible resolutions are also included.

Cluster is not reachable

DataPlane containers are not able to resolve a provided hostname or use the IP address to connect to the machine.

DNS resolution is not setup.

There are firewall or other networking restrictions that are preventing access.

Sample Message:

Failed: This is not a valid Ambari URL.

Procedure

1. Verify that the specified hostname or IP address is valid and reachable from the DP host machine.
2. If the hostname or IP address is reachable, try adding the hostname resolution to the DataPlane container using the ./dpdeploy.sh utils add-host <ip> <host> command.
3. Verify if network connectivity settings, such as firewalls, are configured correctly.

Knox is not set up on the HDP cluster, or Ambari credentials are incorrect for 'seeded user' mode

This error occurs when the cluster is reachable, but authentication is failing.

Sample Message:

Unable to connect, please retry. DataPlane could not retrieve cluster information.

Possible Causes:

- Knox is not set up on the cluster.
- The user wants to use the less secure 'seeded user' mode, but the credentials of the seeded user (user name or password) are not setup in DataPlane.

Procedure

1. Validate that Knox is configured correctly as per documentation.
2. If seeded user mode is being used (for evaluation purposes), add the correct credentials to DataPlane using ./dpdeploy.sh utils update-user ambari.

Knox setup is incorrect on the HDP cluster

This error indicates that the cluster being registered has Knox enabled, but the communication from DataPlane to Knox is failing.

Sample Message:

Failed: There was an error fetching information from Ambari.

Possible Causes:

- The Knox token service is not properly configured.
- The public key of DataPlane is not set up correctly in the Knox topology.

Procedure

Validate that the Knox configuration for the token topology is done correctly, following the instructions in *Configuring Knox and Ranger for DataPlane*.

Cannot register a cluster, other causes

If you cannot register a cluster with DataPlane and none of the errors mentioned are the cause, the following might apply.

Procedure

1. Verify that the hostname where Knox is running is valid and reachable from the DP host machine. If it is reachable, try adding the hostname resolution to the DP container using the ./dpdeploy.sh utils add-host <ip> <host> command.
2. Verify that network connectivity settings, such as firewalls, are correctly configured.
3. Verify that the username is an Ambari Admin on the cluster. If not, make the user an Ambari Admin user, by logging into Ambari, selecting the user, and providing Admin privileges.
4. See the DataPlane Support Matrices and verify that you are running a supported configuration.

Cluster status displays incorrectly on Details page

On the Cluster Details page, sometimes the Status of a cluster displays in gray, instead of red or green.

This generally indicates a timeout issue, in which DataPlane is not able to refresh the cluster details correctly. Manually refreshing the cluster information should fix the problem.

Procedure

1. Click the **Actions** icon at the end of the row.
2. Click **Refresh**.

   Refresh of the cluster status might take several seconds.

Logging in using the DataPlane local admin role

The local admin role allows you to perform administrative activities and troubleshoot problems when access through LDAP and Knox is not available. The local admin is also the role you use to log in to DataPlane the first time, before LDAP is configured in DataPlane for SSO.

About this task

When you log in as the local DataPlane Admin, you bypass Knox.

For login, the default username is “admin”. The password you use to log in is set during the installation process.

Procedure

Log in by appending /sign-in to the DataPlane login URL, for example:

```
http://dataplane-host-name/sign-in
```

wget command is not available

The wget command is not installed on the system.

Use the command yum install wget to install the wget tool.

Delete and clean up Docker containers

If you have problems with your installation or want to update a DataPlane container, you can delete the Docker containers and then install the new images.

About this task

Important: Performing this task deletes all of your DP Platform database content, so you will have to reconfigure the LDAP and cluster registration settings after reinstalling the Docker containers.

For information about the commands and options supported by ./dpdeploy.sh, use the command-line help.

Before you begin

You must be root user to perform this task.

Procedure

1. cd /usr/dp/current/apps/dlm/bin
2. ./dlmdeploy.sh destroy
3. cd /usr/dp/current/core/bin
4. ./dpdeploy.sh destroy --all

5. docker ps

   This ensures that no containers are running. If you see any, kill them with docker kill.
6. Go to *Initialize DataPlane* and run the original DataPlane deployment commands starting with ./dpdeploy.sh init --all.